

7
150pts.

ARUN

Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek





EL SPECTRUM DECIDE



través de los programas propuestos hasta ahora, hemos podido comprobar que el ordenador ejecuta siempre las instrucciones que componen los mismos de una manera secuencial, es decir, una a continuación de otra, respetando su numeración. Este estado de cosas sólo puede ser modificado mediante la sentencia **GO TO**, que obliga a un cambio en la siguiente línea a ejecutar en el programa.

Vamos a ver ahora como nuestro Spectrum es realmente capaz de «tomar decisiones» en base a comparaciones, y seguir por un camino u otro a partir de una evaluación de condiciones. La estructura general de la instrucción a tal fin es

IF condición THEN instrucción

IF, que en inglés quiere decir Si (si condicional), precede a la condición a evaluar, y **THEN**, cuyo traducción es ENTONCES, sirve para determinar el final de la condición e indicar el comienzo de la instrucción, que ha de ejecutarse sólo en caso de que la evaluación haya sido dada por «verdadero». Si, por el contrario, la evaluación de la condición resulta «falsa», pasa a ejecutarse la siguiente línea, haciéndose caso omiso de la instrucción que sigue a **THEN**. El comportamiento de este tipo de instrucciones podríamos describirlo brevemente de la siguiente manera:

Si se cumple determinada condición **ENTONCES** haz lo que sigue **SINO**...

Podemos establecer comparaciones de tipo numérico o de cadena, pero nunca mixtas, es decir, comparar letras con números. Dados dos elementos, podemos realizar las siguientes comparaciones: si son iguales, si el primero es menor que



Las instrucciones **IF** obligan al Spectrum a tomar una decisión entre dos posibles acciones.

el segundo, si el primero es mayor que el segundo, si el primero es menor o igual que el segundo (no es mayor), si el primero es mayor o igual que el segundo (no es menor), y si el primero es diferente del segundo (no es igual). Con esto se agotan todas las combinaciones posibles. Los símbolos que corresponden a estas comparaciones son:

=, <, >, <=, >= y <>



then



Mediante la sentencia **IF...THEN** se lleva a cabo la tarea de decisiones de los ordenadores programados en BASIC.

i!

Las sentencias **GO TO** imponen al ordenador un salto a determinado número de línea, por el simple hecho de pasar el programa por ese punto.

*

Las condiciones condicionales, hacen que el programa se dirija a un futuro número de instrucción, sólo en el caso de que concuerden determinadas circunstancias.

*

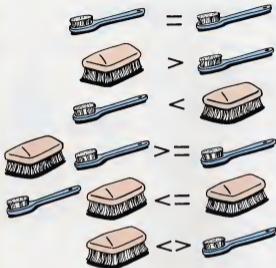
La palabra clave **IF**, precede a la expresión a evaluar, y la palabra clave **THEN**, determina el final de la condición y el principio del comando a ejecutar, si la evaluación de las condiciones resulta verdadera.

*

Las comparaciones deben establecerse siempre entre variables del mismo tipo, es decir, dos números o dos de cadena.

*

Se puede continuar siempre el programa a pesar de que no haya transcurrido la totalidad del tiempo asignado a la sentencia **PAUSE**, pulsando cualquier tecla.



Verán así las posibles comparaciones. Mejor que se puedan resolver en el BASIC de Sinclair.

Como dicho así resulta un poco árido, aplicaremos nuestros conocimientos a algunos ejemplos. Vamos a suponer que, dentro de un programa, los valores de cuatro variables son:

A=12, B=165, A\$="JOSE LUIS" y B\$="JOSE ANGEL"

Y codificamos las siguientes instrucciones:

```
A) 20 IF A=12 THEN PRINT "A ES IGUAL A 12"
B) 20 IF B>A THEN GO TO 90
C) 20 IF A$<"JOSE" THEN GO TO 100
D) 20 IF B$<>A$ THEN PRINT "TU QUIEN ERES? ";A$;" O ";B$
```

En los ejemplos A y B, se establecen compara-

ciones entre elementos de tipo numérico, y en C y D de tipo cadena.

En el primer caso, se comparó la variable **A** con la constante numérica **12**. Como la variable **A** tiene valor **162** en el momento de acceder el programa a la línea 20, no se ejecutará la sentencia **PRINT** que figura a partir de **THEN**, sino que el programa continuará en secuencia con la línea siguiente a la instrucción 20.

En el segundo ejemplo se establece otra comparación de tipo numérico, esta vez entre variables. Atendiendo a los valores de las mismas, como **B** es mayor que **A**, el programa continuará en la línea 90.

En el ejemplo C aparece una comparación entre una variable de cadena y una constante del mismo tipo. En este caso, al comparar el ordenador la cadena "JOSE LUIS" con "JOSE", la primera le resulte mayor que la segunda, puesto que a pesar de comenzar ambas por la misma secuencia de cuatro caracteres, la primera cadena es más larga que la segunda. Debido a este resultado, el



programa continuará ejecutando la siguiente instrucción en secuencia, y no la sentencia de bifurcación a la línea 100.

Finalmente, en el último ejemplo, podemos ver una comparación entre variables de cadena (A9 y B9). En este caso, A9 resulta mayor que B9, aunque no tiene mucha importancia, ya que lo que tratamos de averiguar en esta ocasión es solamente si las variables son diferentes en contenido, por lo que el programa ejecutará el PRINT situado tras THEN.

Debemos considerar que cuando tratamos variables de cadena en comparaciones, se utiliza el ordenamiento interno que el Spectrum tiene para cada uno de los caracteres del teclado (el orden alfabético). De esta forma, la letra «B» se considera menor que la «C», etc.

En las comparaciones con variables de tipo numérico, el ordenador tiene en cuenta en primer lugar los signos y luego los valores. En el caso de que los elementos a comparar sean de diferente signo, resulta siempre mayor el positivo. Si los elementos son del mismo signo, se analiza el valor absoluto (valor sin signo) de las variables. Si las cantidades son positivas, se toma como mayor la de valor absoluto superior, y si son negativas, la de valor absoluto inferior.

En el caso de las cadenas de caracteres, esta operación de comparación se realiza de izquierda a derecha, con el convenio de que si dos cadenas son idénticas en lo que se refiere a sus primeros caracteres, se considera como menor la más corta. De esta forma, la cadena "ABC" se considera siempre menor que la "ABCD".

El juego de caracteres completo del Spectrum puede encontrarse al final de la Obra, aunque normalmente debe bastarnos con saber que los números son para el Spectrum menores que todas las letras, y que las letras mayúsculas son menores que las minúsculas.

LAS DECISIONES SE COMPLICAN

Hasta ahora hemos hablado de las comparaciones más simples, las de tipo unitario, en las que manipulamos un solo par de elementos. Manteniendo la misma estructura, podemos escribir instrucciones mucho más potentes apoyándonos en los OPERADORES RELACIONALES, también llamados OPERADORES LÓGICOS.

Estos operadores, establecen la relación que debe cumplirse entre los diferentes elementos

-4 -3 -2 -1 0 1 2 3 4

En las comparaciones realizadas con datos numéricos, primero se tiene en cuenta el signo y en caso de ser éste igual, se valoran los datos.

que componen la instrucción de comparación, siguiendo el formato

IF condición OPERADOR condición THEN instrucción

En este formato mantenemos una estructura parecida a la de los IF simples, pero haciendo intervenir diferentes condiciones relacionadas con los correspondientes operadores. Los tres tipos de operadores lógicos son **AND**, **OR** y **NOT**, su traducción al castellano es **Y**, **O** y **NO**, respectivamente, su significado lo estudiaremos a continuación mediante unos ejemplos. Comenzaremos hablando del operador **AND**. Su-

No debemos establecer comparaciones entre elementos de distinto tipo (números y cadenas)

CORRECTO



CORRECTO



INCORRECTO



!

Si comparamos dos cadenas que comienzan por lo mismo su cuantía de caracteres se considera menor la más corta.

*

En la comparación entre variables de cadena, se sigue el orden establecido de códigos de caracteres del Spectrum, que podemos encontrar detallado al final de nuestra obra o en el manual del aparato.

*

Al comparar variables de tipo numérico se tiene en cuenta primero su signo y después su valor absoluto.

*

Los operadores lógicos permiten establecer más de una situación a evaluar, para determinar un salto en la secuencia del programa o la ejecución de determinadas instrucciones.

*

Los posibles operadores de comparación son: =, <, >, <=, >= y <>

VERDADERO



FALSO



El operador AND accede que las dos acciones que relaciona sean verdaderas para que la condición se cumpla.

pongamos que tenemos que decidir si vamos o no al cine, dependiendo el hecho de dos condiciones:

1. TENER DINERO PARA LAS ENTRADAS
2. NO HABER VISTO LA PELICULA

Evidentemente, tenemos que relacionar ambas condiciones con el operador AND (Y), ya que las dos deben ser ciertas al mismo tiempo, es decir, no vamos a ir si no tenemos dinero para las entradas o, por supuesto, si la película la hemos visto ya. Por tanto, la decisión por parte del ordenador se tomará por medio de una instrucción que tenga el siguiente significado: Si tenemos dinero suficiente Y no hemos visto la película ENTONCES iremos al cine. Traduciendo la frase anterior al inglés, tendremos anotado gran parte del camino para hacerlo entender a la máquina: IF tenemos dinero suficiente AND no hemos visto la película THEN iremos al cine.

Para terminar de trabajar nuestro problema a un

programa BASIC, sólo nos basta hacer comprender al ordenador las condiciones "tenemos dinero suficiente" y "no hemos visto la película". ¡Menos a la obra! Supongamos que D es el dinero que tenemos, y P9 una película que queremos ver. Vamos a escribir el siguiente programa

```

10 REM IR O NO AL CINE, AC. ACUT. EL DI
110 D=0
120 C=0
130 INPUT "Dinero: " D
140 PRINT "Tenemos " D " Ptas."
150 INPUT "Película: " P9
160 PRINT "y queremos ir a ver?"
170
180 IF D>=500 AND P9<"TRON" THEN
190 GO TO 120
200 PRINT "NO PODEROS IR porque
210
220 IF D<=500 THEN PRINT "NO tiene
230 mos DINERO ?"
240 IF P9<"TRON" THEN PRINT "YA
250 la hemos VISTO?"
260 GO TO 120
270 PRINT "PODEMOS IR SIN PROBLEMA
280
290 PRINT "¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡
300 PAUSE 0
310 GO TO 20
    
```

En la comparación de cadenas se tiene en cuenta el resultado de la comparación lógica entre sus caracteres y, de ser posible, cual de las cadenas comparadas es más larga.

De la ejecución de este programa podemos sacar algunas conclusiones interesantes. Ante todo diremos que como datos fijos (constantes) hemos





introducido el precio de la localidad y una de las películas que ya hemos visto (la única que el programa va a considerar).

Lo deseable para que el programa fuera un práctico consejo a la hora de resolvernos la tarde del fin de semana, sería poder contar con un archivo en la memoria de todas las películas que hemos visto. En el caso de que seamos muy aficionados al séptimo arte, esto podría traernos problemas de falta de memoria, por otro lado, nuestra única intención es que el programa nos sirva de ejemplo de la forma en que debemos codificar la toma de decisiones para el Spectrum. Como ya sabemos, las líneas 10 y 20 se ocupan, respectivamente, de dar información sobre el contenido del programa y de borrar la pantalla. Las líneas 30 a 60 efectúan la entrada de datos del programa y representan la información en la pantalla. Los variables que utilizamos para esta entrada son: D para el dinero de que disponemos y P# para el título de la película candidata. En la línea 70 establecemos la más ardua de las decisiones:

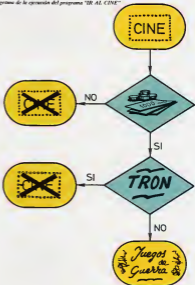
```
70 IF D<<350 AND P#<<"TRON" THEN
GO TO 120
```

De acuerdo con dos condiciones vinculadas por el operador lógico AND, determinamos definitivamente en esta línea si podemos ir o no al cine. Si ambas condiciones resultan ser ciertas (no basta con que lo sea una sola), en base a los datos que introducimos (D y P#), el programa sigue a partir de la línea 120, señalándonos que no existe ningún problema. En caso contrario, el programa continuará en secuencia, haciendo caso omiso de la sentencia GO TO.

Una vez decidido que no podemos ir al cine, entramos en el análisis de los motivos. No existen más que tres posibilidades:

1. No tenemos dinero.
2. Ya hemos visto la película.

Diagrama de la ejecución del programa "IR AL CINE"



El operador OR necesita que uno de los dos (o los dos) factores que se relaciona se cumplan, para que la condición se cumpla.

VERDADERO



FALSO



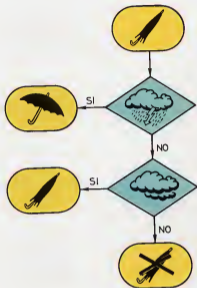


Diagrama de la ejecución del programa "CONDICIONES METEOROLOGICAS"

3. Además de no tener dinero, hemos visto la película.

Este análisis de motivos se concreta en las instrucciones 90 y 100, ya que el simple hecho de que el programa pase por la línea 80, implica que existe al menos una de las causas suficientes para no ir, por lo que imprimimos el mensaje de desesperanza:

```
80 PRINT "NO PODEMOS IR porque "
```

Acto seguido analizamos el porqué hemos llegado a tan triste situación:

```
90 IF D<350 THEN PRINT "NO tenemos DINERO "
```

```
100 IF P#="TRON" THEN PRINT "YA la hemos VISTO"
```

Si tenemos menos de 350 pesetas, se imprimirá el mensaje "NO tenemos DINERO", y si hemos indicado como candidato la película "TRON", lo hará el mensaje de "YA la hemos VISTO". Puede también darse la posibilidad de que concuerden ambas circunstancias, en cuyo caso se imprimirán los dos mensajes.

Entramos por último en el análisis de las líneas 130 a 150, que nos ofrecerán una sorpresa: la aparición en escena por primera vez de la palabra clave PAUSE.

```
130 PRINT "PULSA UNA TECLA"
140 PAUSE 0
150 GO TO 20
```

La sentencia PAUSE se emplea para detener momentáneamente la ejecución del programa, por un espacio de tiempo determinado: el especificado a continuación de la palabra clave. Dicha pausa se expresa en centésimos de segundo, de forma que si deseamos una detención de un segundo deberemos indicar PAUSE 50, si dos segundos PAUSE 100, etc.

En cualquier caso, el programa continuará al pulsar una tecla, a pesar de no haber concluido su tiempo completo de espera. Otra posibilidad es la de provocar una parada en el programa por tiempo indefinido, de la cual la única forma de salir, para que la ejecución continúe, es pulsando una tecla. Esta última opción es la que lleva a cabo como parámetro de la sentencia PAUSE (PAUSE 0).

En nuestro ejemplo, nos servimos de esta instrucción para provocar una pausa indefinida, en la que se muestran los resultados impresos y se solicita que pulsemos una tecla para volver a ejecutar el programa.

De forma similar a la que hemos actuado en el programa propuesto anteriormente, podemos diseñar otro muy parecido en el que emplearemos el operador lógico OR (O). Podría ser un programa en que pediríamos al ordenador que nos avisara si debemos coger la gabardina al salir de casa. Ahora, las condiciones serán:

1. Si llueve
2. Si está nublado.

En esta ocasión, es suficiente que se cumpla cualquiera de ellas para que debamos proveerlos de la indumentaria apropiada, correspondiéndole este caso claramente con la relación OR. Debemos codificar la instrucción 70 de la forma: 70 IF LLUVIA OR MAL TIEMPO THEN GO TO 120. Esta línea sería la traducción al BASIC de la frase «Si llueve O está nublado ENTONCES cogemos la gabardina».

Para facilitar la codificación de la instrucción hemos escogido dos variables cuyo nombre está directamente relacionado con el valor que contienen: LLUVIA y MAL TIEMPO; estas variables son numéricas, de forma que para la entrada de datos en los INPUT, debemos adoptar el convenio de considerar como «1» si el fenómeno atmosférico se produce y como «0» si no. Una novedad en la instrucción 70 es la de que no aparece el símbolo de igualdad detrás de las variables.

LLUVIA=1 y MAL TIEMPO=1

Esto podemos hacerlo gracias a que el ordenador supone que, al no existir el símbolo de igualdad, es nuestro deseo tratar estas variables numéricas como variables lógicas. Las variables lógicas adoptan para la comparación dos únicos valores VERDADERO (diferente de 0) y FALSO (0). Al escribir la instrucción de esta forma, conseguimos que el programa bifurque a la línea 120 cuando LLUVIA o MAL TIEMPO sean, los dos o cualquiera de ellas, diferentes de cero, es decir, verdaderas en sentido lógico.

La codificación del resto del programa será muy simple, puesto que sólo deberemos realizar modificaciones en las sentencias PRINT del programa, según las condiciones meteorológicas que se hayan producido, de manera similar a como procedimos con nuestro programa del capítulo.

Habíamos por último del operador lógico NOT, que es el complementario (negación lógica) de los operadores lógicos que hemos visto hasta ahora. La verdadera utilidad del operador NOT es la de clarificar nuestras intenciones respecto a las condiciones que se evalúan en un programa. Expliquemos esto más claramente, supongamos que determinado suceso depende de una sola condición, por ejemplo, que ir a trabajar depende exclusivamente de que no sea domingo. La sentencia IF que deberíamos utilizar para codificar este suceso sería: SI hoy es diferente de domingo ENTONCES vamos a trabajar.

Para traducir esta instrucción al lenguaje BASIC emplearemos una variable lógica que denominaremos DOMINGO, cuyo valor será cero en caso de que no sea domingo, y uno si hoy es domingo. La instrucción de decisión se podrá construir de la siguiente manera: IF DOMINGO <> 1 THEN vamos a trabajar o IF DOMINGO = 0 THEN vamos a trabajar. En este caso hubiera quedado mucho más clara una decisión del tipo «Si hoy NO es domingo ENTONCES vamos a trabajar», la traducción directa de esta frase al BASIC sería IF NOT DOMINGO THEN vamos a trabajar. De lo dicho, sin duda habremos sacado la siguiente conclusión: NOT variable, tiene exactamente el mismo efecto que variable <> 0. Para terminar, diremos que se pueden construir complicadas estructuras de comprobación de condiciones, utilizando combinaciones de los



La sentencia PAUSE realiza una detención en la ejecución del programa.

operadores lógicos. Del mismo modo que para las operaciones matemáticas, el Spectrum tiene un orden de prioridades de ejecución de estos operadores, de forma que el AND (producto lógico) se ejecuta con prioridad sobre el OR (suma lógica). En cualquier caso, pueden emplearse los paréntesis para alterar el orden en que se ejecutan normalmente las operaciones, de manera similar a lo que ocurre con las operaciones matemáticas. De todas formas, sobre este tema abordaremos próximamente.

Al operador NOT es equivalente en su efecto a la comparación "distinto que".

VERDADERO



= NOT



FALSO



= NOT



!!

Con PAUSE 0, se provoca una parada indefinida de lo que solo se puede salir forzando el teclado para una tecla.

*

La instrucción PAUSE, produce una interrupción en la ejecución del programa, por el espacio de tiempo especificado como argumento de la sentencia ejecutada en cualquier momento de ejecución de seguridad.

EL T.O.S.



El disco **INVESTORISK 200** se controla gracias al sistema operativo TOS (Telex Operating System), veremos a continuación una breve descripción de sus comandos fundamentales.

Estos comandos son una extensión del BASIC del Spectrum y pueden ser utilizados dentro de un programa o bien como comandos directos, siempre con la precaución de marcar las órdenes que al TOS se refieren, con un asterisco al final de las mismas. Es decir, los comandos utilizados por el TOS son los mismos que los del Spectrum, pero marcados con el asterisco (aunque algunos no realizarán la misma función).

Cuando almacenamos información mediante un cassette, no hay manera rápida de saber lo que tenemos grabado, ni lo que está ocupado en una determinada cinta, o no ser que lo hayamos anotado cuidadosamente. En la unidad de disco, este problema se resuelve fácilmente con el uso de **CAT ***. Este comando nos va a informar, mediante una impresión en la pantalla, de todos los programas que se encuentran en la cara del disco con la que estamos trabajando. Proporciona además los siguientes datos: la longitud de cada uno de ellos, el número de Kbytes ocupados redondeados al entero superior más próximo, si están protegidos o no, el número de Kbytes que todavía quedan libres y los utilizados y, cuando se trata de un fichero, si está abierto o no.

NOMBRES DE PROGRAMAS Y FICHEROS

BITS

En algunas ocasiones, por ejemplo durante la ejecución de un **INPUT**, puede desaparecer el cursor cuando se presiona la tecla que genera la información de nuestro programa o incluso desparece a un punto al rotar, para ello basta con incluir la siguiente instrucción **POKE 23469,224**.



Si alguno todavía duda de los sobornos que puede provocar como el Spectrum que teclee **POKE 23469,1** (transmitir pulso **ENTER**).

Podemos utilizar cualquier nombre que deseemos con las siguientes limitaciones:

* No tendrán una longitud mayor que ocho caracteres, opcionalmente, a continuación del nombre y separados por un punto (.) se pueden incluir otros tres caracteres, que nos sirven para diferenciar tipos de programas, estos apóditos de los nombres se denominan extensiones. (ext abreviadamente). Por ejemplo, se utiliza **.BAS** en programas realizados en BASIC .DAT para bloques de datos. Pero podemos emplear cualquiera tres caracteres que consideremos convenientes.

* Algunos caracteres como **↑**, **+**, o **?**, no se pueden incluir en el nombre, por estar reservados para el sistema operativo; la inclusión de mayúsculas o minúsculas es indiferente (el TOS convertirá todos los caracteres alfabéticos en mayúsculas).

El disco es un dato la mejor de las soluciones para convertir el Spectrum en un sistema de gestión.



El sistema operativo puede organizar los archivos en forma jerárquica, es decir, como si se tratara de un árbol invertido, con la raíz en la parte superior y las ramas hacia abajo. Si al ejecutar el comando **CAT *** encontramos ficheros con la extensión **.DIR**, esto indica que tenemos un directorio dentro de otro en alguna de las distintas ramas del árbol. El TOS permite la creación de dichas ficheros de direcciones en cada cara del disco y de ocho niveles distintos en los que andar los archivos. Se puede acceder a las ramas del árbol mediante los comandos **GOSUB *** o **GOTO *** "nombre DIR".

Para la carga desde el disco se utiliza la sentencia **LOAD**, cuyo sintaxis es: **LOAD *** "nombre" o, si hemos hecho uso de los tres caracteres op-

SENTENCIAS INCORPORADAS POR EL TOS

ALTER *	INPUT *
CAT *	LIST *
CLOSE # *	LOAD *
MOVE *	MERGE *
DIM *	LET *
ERASE *	OPEN # *
FORMAT *	PRINT *
GOSUB *	DRAW *
GOTO *	SAVE *

cionales de extensión, **LOAD *** "nombre.ext". Su empleo es muy similar a la sentencia homónima para el casete. Es importante tener en cuenta que con el disco no podremos cargar ficheros sin indicar el nombre, tal como hacemos con el casete, por tanto, el TOS no identificará el orden **LOAD *** "". Siempre hará falta especificar el nombre del programa. Esto no produce ningún contratiempo pues, aunque no recordemos el nombre exacto, va a ser fácil averiguarlo utilizando el comando **CAT ***.

SAVE * es el comando inverso a **LOAD ***. Hay que destacar que cuando grabemos un programa o un fichero al que queremos dar un nombre que ya figura en el directorio del disco, el TOS nos advertirá de esta circunstancia mediante el mensaje:

"nombre del programa" already exists
Suprese de (Y/N)?

Si nuestra respuesta es **Y**, borrará el programa existente ya mencionado y grabará la nueva versión desde la memoria. Si por el contrario pulsamos **N** ignorará la operación **SAVE ***.

De forma similar al casete o al ZX Microdrive podemos utilizar **SAVE *** "nombre.ext" LINE xxxx, que tiene el efecto de grabar nuestro programa con autoejecución a partir de la línea xxxx. También podremos utilizar los sufixos **SCREEN**, **CODE** o **DATA** para almacenar pantallas, bloques de memoria o dimensionados, respectivamente.

El funcionamiento de **MERGE *** es también análogo al correspondiente comando del BASIC. Simplemente estándar mezcla un programa nuevo y sus variables, con el que se encuentra presente en



Recordemos que los dos tipos fundamentales de ficheros se denominan **SECUENCIALES** y **ALEATORIOS**. Estos se diferencian por el modo de acceso a cada uno de sus campos.



Para el acceso al cualquier campo de un fichero secuencial es necesario leer previamente todos los campos anteriores a él.



El acceso a un fichero aleatorio se realiza directamente a cualquiera de sus campos, dando número por su número de orden dentro del fichero, de forma similar a los dimensionados del BASIC.

BITS

El espacio total de memoria disponible, en electros el dificultad o gráficos de video por el usuario es fácil de cambiar mediante una simple llamada a la ROM del sistema. En cualquier momento podemos ejecutar la siguiente instrucción **PRINT 46536-USH 7562," Bytes libres"**. Si el Spectrum es de 16 K, sustituiremos el 65536 por 32768.



Incluir sonidos dentro de nuestros programas nos hace más interesante. A continuación se listan tres efectos sencillos que podremos aplicar muy fácilmente. Cada **10 FOR N=0 TO 60 STEP 5: BEEP 0.3,N: NEXT N**
Sirena **10 BEEP 36.24: BEEP 36.12: GOTO 10**
«Click-Click» **10 FOR N=0 TO 200: BEEP 0.1,20: PAUSE 10: NEXT N**



Si queremos asegurarnos desde dentro de un programa, que una lista de datos es realmente en mayúsculas, nuestro problema se reduce a ejecutar la siguiente instrucción **POKE 23666,8**. Si por el contrario lo que queremos es convertir el modo minúsculas, sólo tenemos que cambiar el valor del **POKE** a cero.

la memoria del Spectrum. Lógicamente, no es posible realizar un **MERGE** con un bloque de bytes o datos, sino sólo con programas. La sintaxis de la instrucción es: **MERGE "nombre.ext"**.

El comando **FORMAT** es probablemente el más importante de todos, puesto que es obligatorio su ejecución previamente a la utilización de cualquier disquete virgen.

Esta es la única sentencia del TOS en la que se debe especificar el nombre de la unidad de disco: con la que vamos a trabajar. Para la identificación del drive (unidad de disco) afectada, utilizamos las primeras letras del alfabeto, asignándosele la **A** a la primera unidad, **B** a la segunda, **C** a la tercera y **D** a la cuarta. La sintaxis completa de la instrucción es: **FORMAT "nombre de la unidad" "TQ" "nombre del disquete"**. Esta sentencia se introduce por el teclado como comando directo.

Cuando se ejecuta el comando **FORMAT**, el sistema operativo divide la superficie del disquete insertado en la unidad de disco en cuarenta pistas de cuatro Kbytes, y cada pista en sectores. Es decir, lo prepara para que la información que irá a ser almacenada, se distribuya ordenadamente y luego pueda tener fácil acceso a ella. El «formatado» de un disco dura poco más de treinta segundos, y el sistema operativo que se

El disquete de demostración que se suministra con la unidad de disco, sólo va a ser utilizado en un sistema de disco.

SECUENCIAL



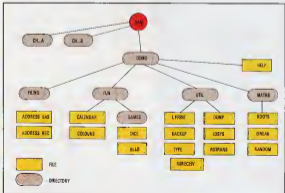
ALEATORIO



El acceso a los puffs de un tubo es secuencial por el momento, de haberse dispuesto en una caja, podríamos hablar de acceso aleatorio.

ocupa en el disquete ocupa 16 K. Otras 4 K se utilizan para el directorio e catálogo del mismo, de ahí que de las bobinas 160 K de cada cara del disquete, sólo podamos aprovechar 140 K.

Es importante tener cuidado con este comando, pues al ejecutarlo se destruye totalmente el contenido del disquete. De todos modos, el TOS se encarga de que queramos «formatar» el disquete



to, realizando una pregunta para confirmarlo antes de ejecutar el comando. Así mismo, cuando sólo disponemos de un drive, el TOS nos irá informando de cuando debemos retirar el disquete a «formatear» para insertar el del sistema operativo.
El comando para el borrado de ficheros es **ERASE** * y su sintaxis correcta es **ERASE** * "nombre.ext". Antes de ejecutarlo, el sistema operativo realizará una pregunta de confirmación de la forma:

ERASE "nombre.ext" Y/N?

Para que la sentencia **ERASE** * tenga efecto es necesario que el fichero que queremos eliminar no se encuentre protegido. Sabremos si lo está ejecutando **CAT** * y mirando en la columna marcada con **P**, si encontramos una letra **P** en la línea correspondiente al fichero a borrar, esto será indicativo de que éste se encuentra protegido. Uno de los comandos del TOS que en BASIC Sindar estándar tienen un significado absolutamente diferente es **ATTR** *. Este comando permite proteger, desproteger, hacer invisible o visible un determinado fichero del directorio. Su sintaxis es:

ATTR * "nombre.ext"p - protege el fichero
ATTR * "nombre.ext"i - hace invisible al fichero
ATTR * "nombre.ext"u - desprotege el fichero
ATTR * "nombre.ext"v - visible el fichero

El que un fichero o programa se haga invisible mediante el uso de **ATTR** *, solamente significa que no aparecerá en el directorio cuando ejecutemos **CAT** *; esta medida es similar a la adoptada por el ZX Microdrive cuando el nombre de un fichero comienza por el carácter caro (CHR\$ 0).

OTRAS PARTICULARIDADES DEL TOS

Una de las posibilidades que ofrece el TOS es la de incluir en el nombre del programa o fichero, los caracteres + o ? cuando usamos determinados comandos. Así por ejemplo, si ejecutamos **CAT** * "+BAS", obtendremos un directorio con el nombre de todas las ficheros de extensión .BAS; con **CAT** * "+?", obtendremos las ficheros a los que no hemos dado extensión, y **CAT** * "R???????" +", buscará una lista de los archivos



Las sentencias básicas utilizadas por el TOS para el control del disco, son muy similares a las empleadas por el ZX INTERFACE 1, pero la gestión del Microdrive

o programas que comienzan por R. Estos caracteres se pueden utilizar con los comandos **ERASE** *, **ATTR** *, **MOVE** *, **LET** * y **CAT** *. Mediante la instrucción **LET** * "nombre antiguo.ext" TO "nuevo nombre.ext", conseguiremos cambiar el nombre de un fichero por otro nuevo. Continuando con las alteraciones de denominación, **GOSUB** * "nombre de la unidad" d se utiliza para cambiar la unidad de disco a que se hace referencia, donde el nombre de la unidad se designa con las letras A, B, C o D.

El comando **CAT** * proporciona una rápida información sobre el contenido de un disquete



BITS

Este sencillo programa proporciona dibujos sencillos en la pantalla, usando diferentes y sin el lenguaje del desagradable lenguaje Integer out of range.

```

10 LET X=RND*
255 LET Y=RND*175
INR RND*? PLOT XY
20 LET A=RND*
255-X LET B=RND*
175-Y DRAW A,D
DRAW D,B
30 GOTO 10

```

Prueba ahora eliminando la línea 20



Una vez realizada la operación de formateado, el disquete tiene 100 Kbytes netos para el usuario por cada cara.

Para finalizar con este breve repaso al TOS, mencionaremos que la copia de programas se realiza mediante `MOVE * "nombre1 ext" TO "nombre2 ext"`. La potencia del sistema opera-

El sistema INVESTISK 200 es incompatible con el ZX INTERFACE 1, y por tanto no permite la utilización sencilla de Microdrives.



tro del INVESTISK es aún mayor que lo aquí expuesto. Se dispone de un total de dieciocho comandos destinados al disco, gracias a los cuales éste se convierte en un periférico muy útil y eficaz. Por último cabe añadir que el apoyo que presta el TOS al programador, va desde las preguntas de confirmación ante operaciones tan delicadas como el `FORMAT` o el borrado, hasta la emisión de un total de cuarenta y seis tipos diferentes de mensajes de error, todos ellos detallados en el manual de operación que acompaña a la unidad de disco.

FICHEROS DE DATOS

La principal ventaja de utilizar la unidad de discos frente a otros sistemas de almacenamiento, es la posibilidad de manejar ficheros de acceso aleatorio (en el ZX Microdrive, como con algunos sistemas de cinta, sólo se pueden manejar ficheros secuenciales).

Cada uno de los componentes de un fichero se denomina `RECORD` o `CAMPO` (en ocasiones `REGISTRO`), la forma de acceder a cada uno de los campos de un fichero, es la que marca el tipo de éste. Fundamentalmente, existen dos tipos de ficheros: `SECUENCIALES` y `ALEATORIOS`.

Los primeros se caracterizan porque para acceder a cualquiera de sus campos es necesario haber leído previamente todos los anteriores. Imaginemos un tubo de pastillas dulces, si aquello cuyo sabor estamos buscando se encuentra al final del tubo, nos veremos obligados a sacar todas las pastillas anteriores para llegar hasta la que buscamos. Este es el tipo de acceso secuencial, utilizado en el ZX Microdrive.

Cuando podemos acceder a cada uno de los records de un fichero sin pasar por los anteriores, decimos que este fichero es de acceso aleatorio. Siguiendo el ejemplo anterior, podríamos imaginar que las pastillas han sido distribuidas en una caja formando un solo piso, gracias a ello podremos llegar directamente a cada una de ellas sin necesidad de desplazar el resto. Este tipo de ficheros es sin duda el más conveniente puesto que, aunque más lentamente, simula el tipo de acceso que en el BASIC se tiene a los dimensionados.

Para el control de ambos tipos de ficheros, el TOS emplea las sentencias `OPEN # *, CLOSE # *, INPUT * y PRINT *`. Cuando se trata de ficheros aleatorios, el `record #` a acceder dentro del fichero se señala a continuación del símbolo `AT`. 

AVENTURA INTERMINABLE

ENTRE los programas que gozan de mayor aceptación dentro de los destinados a microordenadores, se cuentan los del tipo AVENTURA. Gracias a ellos nos trasladamos a un mundo real, en el que el destino de un personaje y el final de una historia depende directamente de nosotros, de nuestras decisiones y comportamiento. Este argumento, muy similar al de la conocida película de fantasía «La Historia Interminable», tiene también brillantes exponentes en el software destinado a nuestro aparato. Tal es el caso de programas tan conocidos como «El Hobbit» (Melbourne House) u otros muchos. Sin embargo, no debemos pensar que la realización de este tipo de programas nos está vedada a los desconocedores del código máquina. También un BASIC se pueden conseguir algunos «aventuras» interesantes.

Sierva como botón de muestra el programa que figura a continuación, aunque hemos de tener en cuenta que se trata de una versión reducida, puesto que de no ser así, la gran ocupación de espacio haría imposible su publicación.

LA HISTORIA

Este juego aventura nos permitirá explorar nuevos y extraños países, encontrar tesoros, luchar contra monstruos, tendremos que sobrevivir en un inhóspito y engañoso desierto. En pocas palabras si queremos acabar con vida, hemos de resolver todos y cada uno de los problemas que se nos plantearán en el nuevo mundo al que vamos a entrar. Obviamente, nosotros seremos los protagonistas de esta «historia interminable».

El Spectrum nos describirá los lugares que vayamos visitando y los objetos que aparezcan. Comenzaremos en un deteriorado lugar del condado de Karkovia, perteneciente al territorio de nuestro gran enemigo ASHAMM, el dragón verde.

Nuestro objetivo es hallar los tesoros de la prin-



cipal Sireka y depositarlos en el punto de partida. Para ello tendremos que hacer uso adecuado, en el momento oportuno, de los objetos que encontremos a nuestro paso, y con ellos salir arosos de los múltiples problemas con los que nos vamos a topar. Pero ¡CUIDADO!, los sergentitos parias del país vecino están al acecho. Hemos de tener en cuenta, que todos los objetos que nos encontremos, por extraños que nos parezcan, tienen una utilidad y puede que nos sean necesarios en algún momento de la aventura. Veremos un ejemplo a modo de pista: si intentamos abrirnos el lago del caso del desierto, con proveernos de un tubo hueco que poder utilizar para respirar mientras andamos por el fondo, pereceremos en el intento.

Lamentablemente, no podremos portar todos los objetos que nos vayamos encontrando, por lo que deberemos seleccionar con cuidado nuestro equipaje. Ante todo hemos de utilizar la imaginación, tengamos en cuenta que somos un personaje de fantasía que ¡hasta puede comer lágrimas! Sin embargo, tenemos en lo posible nuestra hambre pantagruélica, puesto que esa misma lágrima nos puede servir para ir hacia la parte norte de la casa en que comienza nuestra aventura, donde reina la oscuridad; al intentar entrar allí, siem-

!!

Los subrutinas han sido situadas al comienzo del programa, para facilitar un acceso más rápido a los mismos.

*

Los programas del tipo AVENTURA no han de estar necesariamente escritos en código máquina; también en BASIC podéis conseguir resultados bastante aceptables.

*

El factor atencioso influye en el programa, gracias a lo cual cada nueva aventura será diferente a las anteriores.

pre que no vayamos provistos de la lámpara, el programa emitirá el mensaje "ALGO TE FRENA" (¡o oscuridad!), no permitiéndonos el acceso al lugar.

Bueno, ya basta de pistas. Antes de meternos de lleno en el estudio del manejo del programa, comentemos un último consejo: los programas de aventura precisan de mucha paciencia e imaginación. No hay que desesperarse por tanto si las duntas del «desierto ibérico», en su incansable viaje, bloquean nuestros posibles salidas. En último caso siempre podemos interrumpir el programa y recomenzar con RUN. Recordad: «No es mejor pescador el que tiene la caña más larga, sino el que tiene la paciencia más grande». Los comandos a utilizar para «actuar» en nuestra historia consisten en una o dos palabras. Habrá que teclear en primer lugar el verbo (es decir, la acción que deseamos realizar), y a continuación (o siempre) nombrar el objeto con el que vamos

a realizar dicha acción. En algunos casos el ordenador nos pedirá más información después de haber dado una orden, por ejemplo si deseamos "MATA DRAGON", el Spectrum nos preguntará con que deseamos destruir a nuestro acérrimo enemigo; en este caso hay que introducir el nombre del arma o herramienta con que deseamos hacerlo, o nada si queremos utilizar nuestras propias manos.

Ante todo hay que introducir todos los comandos correctamente, esto es, bien deletreados. Si alguna letra de la orden a ejecutar estuviese equivocada, el ordenador hará caso omiso, y te presentará en la pantalla el mensaje "NO COMPRENDO", seguido del comando erróneo introducido.

Para ahorrarnos trabajo y evitar errores de escritura, la mayoría de las palabras-comando pueden ser abreviadas a 1, 2 ó 3 letras iniciales. Por otra parte, es conveniente dibujar un mapa de nues-





tro recorrido, para tomar una referencia del punto en donde nos encontramos, evitando así nuestra desorientación.

Como pista, y sin que sirva de precedente, diremos que existe un oasis al sureste de la posición inicial, y que las dunas del desierto pueden bloquear las salidas que hubicamos descubriendo con anterioridad. En la tabla adjunta se detallan todos los comandos que el ordenador reconoce. Con suerte, paciencia e imaginación, y valiéndose sólo de las palabras comando reseñados más adelante, podremos llegar a conseguir nuestro objetivo.

La información que el ordenador necesita de cada objeto consiste en su propia descripción (donde está, qué clase de objeto es (tesoro, monstruo, llave...), cómo es de fuerte, etc.). Este sistema permite una mayor flexibilidad para la actuación del programa, según la actitud que el protagonista adopte ante el objeto en cuestión.

Tanto los objetos como los lugares y comandos, se encuentran codificados en el programa mediante un número, que supone un diferente significado en cada caso. Cuando se trata de objetos, éstos pueden poseer tres valores diferentes:

- * valor -1: objetos que no se pueden ver o usar, u objetos cogidos con anterioridad.

- * valor 0: objetos que el jugador lleva consigo.

- * valor 1: resto de los objetos que no son utilizados en ese momento por la situación del jugador.

Los números que han sido asignados a los verbos (coge, deja, come...) son utilizados por el programa para localizar la subrutina que realiza la acción de dicho verbo.

La codificación de cada comando se corresponde con la posición que ostenta cada uno de los **DATA**. Así pues, el primer verbo llevará el número uno, y la subrutina de acceso se encontrará a partir de la línea de programa número **1463 + 31 *** (Número del verbo). Ya sólo resta añadir que el programa tiene asignados unos números claves, los cuales son empleados por el ordenador para localizar el lugar o escenario en el que discurre la acción presente, estos valores comienzan en el uno y van incrementándose hasta llegar al treinta.

```

1 REM *****
2 REM * 2.8-ARCADE, 02/09/84 *
3 REM *****
4
5 GO SUB 1463
6
7 REM
8
9 REM
10 REM
11 REM
12 REM
13 CLR
14 GO SUB 1463
15 GO SUB 1463
16 REM *****
17 REM *****
18 REM *****
19 REM *****
20 REM *****
21 REM *****
22 REM *****
23 REM *****
24 REM *****
25 REM *****
26 REM *****
27 REM *****
28 REM *****
29 REM *****
30 REM *****
31 REM *****
32 REM *****
33 REM *****
34 REM *****
35 REM *****
36 REM *****
37 REM *****
38 REM *****
39 REM *****
40 REM *****
41 REM *****
42 REM *****
43 REM *****
44 REM *****
45 REM *****
46 REM *****
47 REM *****
48 REM *****
49 REM *****
50 REM *****
51 REM *****
52 REM *****
53 REM *****
54 REM *****
55 REM *****
56 REM *****
57 REM *****
58 REM *****
59 REM *****
60 REM *****
61 REM *****
62 REM *****
63 REM *****
64 REM *****
65 REM *****
66 REM *****
67 REM *****
68 REM *****
69 REM *****
70 REM *****
71 REM *****
72 REM *****
73 REM *****
74 REM *****
75 REM *****
76 REM *****
77 REM *****
78 REM *****
79 REM *****
80 REM *****
81 REM *****
82 REM *****
83 REM *****
84 REM *****
85 REM *****
86 REM *****
87 REM *****
88 REM *****
89 REM *****
90 REM *****
91 REM *****
92 REM *****
93 REM *****
94 REM *****
95 REM *****
96 REM *****
97 REM *****
98 REM *****
99 REM *****
100 REM *****
101 REM *****
102 REM *****
103 REM *****
104 REM *****
105 REM *****
106 REM *****
107 REM *****
108 REM *****
109 REM *****
110 REM *****
111 REM *****
112 REM *****
113 REM *****
114 REM *****
115 REM *****
116 REM *****
117 REM *****
118 REM *****
119 REM *****
120 REM *****
121 REM *****
122 REM *****
123 REM *****
124 REM *****
125 REM *****
126 REM *****
127 REM *****
128 REM *****
129 REM *****
130 REM *****
131 REM *****
132 REM *****
133 REM *****
134 REM *****
135 REM *****
136 REM *****
137 REM *****
138 REM *****
139 REM *****
140 REM *****
141 REM *****
142 REM *****
143 REM *****
144 REM *****
145 REM *****
146 REM *****
147 REM *****
148 REM *****
149 REM *****
150 REM *****
151 REM *****
152 REM *****
153 REM *****
154 REM *****
155 REM *****
156 REM *****
157 REM *****
158 REM *****
159 REM *****
160 REM *****
161 REM *****
162 REM *****
163 REM *****
164 REM *****
165 REM *****
166 REM *****
167 REM *****
168 REM *****
169 REM *****
170 REM *****
171 REM *****
172 REM *****
173 REM *****
174 REM *****
175 REM *****
176 REM *****
177 REM *****
178 REM *****
179 REM *****
180 REM *****
181 REM *****
182 REM *****
183 REM *****
184 REM *****
185 REM *****
186 REM *****
187 REM *****
188 REM *****
189 REM *****
190 REM *****
191 REM *****
192 REM *****
193 REM *****
194 REM *****
195 REM *****
196 REM *****
197 REM *****
198 REM *****
199 REM *****
200 REM *****
201 REM *****
202 REM *****
203 REM *****
204 REM *****
205 REM *****
206 REM *****
207 REM *****
208 REM *****
209 REM *****
210 REM *****
211 REM *****
212 REM *****
213 REM *****
214 REM *****
215 REM *****
216 REM *****
217 REM *****
218 REM *****
219 REM *****
220 REM *****
221 REM *****
222 REM *****
223 REM *****
224 REM *****
225 REM *****
226 REM *****
227 REM *****
228 REM *****
229 REM *****
230 REM *****
231 REM *****
232 REM *****
233 REM *****
234 REM *****
235 REM *****
236 REM *****
237 REM *****
238 REM *****
239 REM *****
240 REM *****
241 REM *****
242 REM *****
243 REM *****
244 REM *****
245 REM *****
246 REM *****
247 REM *****
248 REM *****
249 REM *****
250 REM *****
251 REM *****
252 REM *****
253 REM *****
254 REM *****
255 REM *****
256 REM *****
257 REM *****
258 REM *****
259 REM *****
260 REM *****
261 REM *****
262 REM *****
263 REM *****
264 REM *****
265 REM *****
266 REM *****
267 REM *****
268 REM *****
269 REM *****
270 REM *****
271 REM *****
272 REM *****
273 REM *****
274 REM *****
275 REM *****
276 REM *****
277 REM *****
278 REM *****
279 REM *****
280 REM *****
281 REM *****
282 REM *****
283 REM *****
284 REM *****
285 REM *****
286 REM *****
287 REM *****
288 REM *****
289 REM *****
290 REM *****
291 REM *****
292 REM *****
293 REM *****
294 REM *****
295 REM *****
296 REM *****
297 REM *****
298 REM *****
299 REM *****
300 REM *****
301 REM *****
302 REM *****
303 REM *****
304 REM *****
305 REM *****
306 REM *****
307 REM *****
308 REM *****
309 REM *****
310 REM *****
311 REM *****
312 REM *****
313 REM *****
314 REM *****
315 REM *****
316 REM *****
317 REM *****
318 REM *****
319 REM *****
320 REM *****
321 REM *****
322 REM *****
323 REM *****
324 REM *****
325 REM *****
326 REM *****
327 REM *****
328 REM *****
329 REM *****
330 REM *****
331 REM *****
332 REM *****
333 REM *****
334 REM *****
335 REM *****
336 REM *****
337 REM *****
338 REM *****
339 REM *****
340 REM *****
341 REM *****
342 REM *****
343 REM *****
344 REM *****
345 REM *****
346 REM *****
347 REM *****
348 REM *****
349 REM *****
350 REM *****
351 REM *****
352 REM *****
353 REM *****
354 REM *****
355 REM *****
356 REM *****
357 REM *****
358 REM *****
359 REM *****
360 REM *****
361 REM *****
362 REM *****
363 REM *****
364 REM *****
365 REM *****
366 REM *****
367 REM *****
368 REM *****
369 REM *****
370 REM *****
371 REM *****
372 REM *****
373 REM *****
374 REM *****
375 REM *****
376 REM *****
377 REM *****
378 REM *****
379 REM *****
380 REM *****
381 REM *****
382 REM *****
383 REM *****
384 REM *****
385 REM *****
386 REM *****
387 REM *****
388 REM *****
389 REM *****
390 REM *****
391 REM *****
392 REM *****
393 REM *****
394 REM *****
395 REM *****
396 REM *****
397 REM *****
398 REM *****
399 REM *****
400 REM *****
401 REM *****
402 REM *****
403 REM *****
404 REM *****
405 REM *****
406 REM *****
407 REM *****
408 REM *****
409 REM *****
410 REM *****
411 REM *****
412 REM *****
413 REM *****
414 REM *****
415 REM *****
416 REM *****
417 REM *****
418 REM *****
419 REM *****
420 REM *****
421 REM *****
422 REM *****
423 REM *****
424 REM *****
425 REM *****
426 REM *****
427 REM *****
428 REM *****
429 REM *****
430 REM *****
431 REM *****
432 REM *****
433 REM *****
434 REM *****
435 REM *****
436 REM *****
437 REM *****
438 REM *****
439 REM *****
440 REM *****
441 REM *****
442 REM *****
443 REM *****
444 REM *****
445 REM *****
446 REM *****
447 REM *****
448 REM *****
449 REM *****
450 REM *****
451 REM *****
452 REM *****
453 REM *****
454 REM *****
455 REM *****
456 REM *****
457 REM *****
458 REM *****
459 REM *****
460 REM *****
461 REM *****
462 REM *****
463 REM *****
464 REM *****
465 REM *****
466 REM *****
467 REM *****
468 REM *****
469 REM *****
470 REM *****
471 REM *****
472 REM *****
473 REM *****
474 REM *****
475 REM *****
476 REM *****
477 REM *****
478 REM *****
479 REM *****
480 REM *****
481 REM *****
482 REM *****
483 REM *****
484 REM *****
485 REM *****
486 REM *****
487 REM *****
488 REM *****
489 REM *****
490 REM *****
491 REM *****
492 REM *****
493 REM *****
494 REM *****
495 REM *****
496 REM *****
497 REM *****
498 REM *****
499 REM *****
500 REM *****
501 REM *****
502 REM *****
503 REM *****
504 REM *****
505 REM *****
506 REM *****
507 REM *****
508 REM *****
509 REM *****
510 REM *****
511 REM *****
512 REM *****
513 REM *****
514 REM *****
515 REM *****
516 REM *****
517 REM *****
518 REM *****
519 REM *****
520 REM *****
521 REM *****
522 REM *****
523 REM *****
524 REM *****
525 REM *****
526 REM *****
527 REM *****
528 REM *****
529 REM *****
530 REM *****
531 REM *****
532 REM *****
533 REM *****
534 REM *****
535 REM *****
536 REM *****
537 REM *****
538 REM *****
539 REM *****
540 REM *****
541 REM *****
542 REM *****
543 REM *****
544 REM *****
545 REM *****
546 REM *****
547 REM *****
548 REM *****
549 REM *****
550 REM *****
551 REM *****
552 REM *****
553 REM *****
554 REM *****
555 REM *****
556 REM *****
557 REM *****
558 REM *****
559 REM *****
560 REM *****
561 REM *****
562 REM *****
563 REM *****
564 REM *****
565 REM *****
566 REM *****
567 REM *****
568 REM *****
569 REM *****
570 REM *****
571 REM *****
572 REM *****
573 REM *****
574 REM *****
575 REM *****
576 REM *****
577 REM *****
578 REM *****
579 REM *****
580 REM *****
581 REM *****
582 REM *****
583 REM *****
584 REM *****
585 REM *****
586 REM *****
587 REM *****
588 REM *****
589 REM *****
590 REM *****
591 REM *****
592 REM *****
593 REM *****
594 REM *****
595 REM *****
596 REM *****
597 REM *****
598 REM *****
599 REM *****
600 REM *****
601 REM *****
602 REM *****
603 REM *****
604 REM *****
605 REM *****
606 REM *****
607 REM *****
608 REM *****
609 REM *****
610 REM *****
611 REM *****
612 REM *****
613 REM *****
614 REM *****
615 REM *****
616 REM *****
617 REM *****
618 REM *****
619 REM *****
620 REM *****
621 REM *****
622 REM *****
623 REM *****
624 REM *****
625 REM *****
626 REM *****
627 REM *****
628 REM *****
629 REM *****
630 REM *****
631 REM *****
632 REM *****
633 REM *****
634 REM *****
635 REM *****
636 REM *****
637 REM *****
638 REM *****
639 REM *****
640 REM *****
641 REM *****
642 REM *****
643 REM *****
644 REM *****
645 REM *****
646 REM *****
647 REM *****
648 REM *****
649 REM *****
650 REM *****
651 REM *****
652 REM *****
653 REM *****
654 REM *****
655 REM *****
656 REM *****
657 REM *****
658 REM *****
659 REM *****
660 REM *****
661 REM *****
662 REM *****
663 REM *****
664 REM *****
665 REM *****
666 REM *****
667 REM *****
668 REM *****
669 REM *****
670 REM *****
671 REM *****
672 REM *****
673 REM *****
674 REM *****
675 REM *****
676 REM *****
677 REM *****
678 REM *****
679 REM *****
680 REM *****
681 REM *****
682 REM *****
683 REM *****
684 REM *****
685 REM *****
686 REM *****
687 REM *****
688 REM *****
689 REM *****
690 REM *****
691 REM *****
692 REM *****
693 REM *****
694 REM *****
695 REM *****
696 REM *****
697 REM *****
698 REM *****
699 REM *****
700 REM *****
701 REM *****
702 REM *****
703 REM *****
704 REM *****
705 REM *****
706 REM *****
707 REM *****
708 REM *****
709 REM *****
710 REM *****
711 REM *****
712 REM *****
713 REM *****
714 REM *****
715 REM *****
716 REM *****
717 REM *****
718 REM *****
719 REM *****
720 REM *****
721 REM *****
722 REM *****
723 REM *****
724 REM *****
725 REM *****
726 REM *****
727 REM *****
728 REM *****
729 REM *****
730 REM *****
731 REM *****
732 REM *****
733 REM *****
734 REM *****
735 REM *****
736 REM *****
737 REM *****
738 REM *****
739 REM *****
740 REM *****
741 REM *****
742 REM *****
743 REM *****
744 REM *****
745 REM *****
746 REM *****
747 REM *****
748 REM *****
749 REM *****
750 REM *****
751 REM *****
752 REM *****
753 REM *****
754 REM *****
755 REM *****
756 REM *****
757 REM *****
758 REM *****
759 REM *****
760 REM *****
761 REM *****
762 REM *****
763 REM *****
764 REM *****
765 REM *****
766 REM *****
767 REM *****
768 REM *****
769 REM *****
770 REM *****
771 REM *****
772 REM *****
773 REM *****
774 REM *****
775 REM *****
776 REM *****
777 REM *****
778 REM *****
779 REM *****
780 REM *****
781 REM *****
782 REM *****
783 REM *****
784 REM *****
785 REM *****
786 REM *****
787 REM *****
788 REM *****
789 REM *****
790 REM *****
791 REM *****
792 REM *****
793 REM *****
794 REM *****
795 REM *****
796 REM *****
797 REM *****
798 REM *****
799 REM *****
800 REM *****
801 REM *****
802 REM *****
803 REM *****
804 REM *****
805 REM *****
806 REM *****
807 REM *****
808 REM *****
809 REM *****
810 REM *****
811 REM *****
812 REM *****
813 REM *****
814 REM *****
815 REM *****
816 REM *****
817 REM *****
818 REM *****
819 REM *****
820 REM *****
821 REM *****
822 REM *****
823 REM *****
824 REM *****
825 REM *****
826 REM *****
827 REM *****
828 REM *****
829 REM *****
830 REM *****
831 REM *****
832 REM *****
833 REM *****
834 REM *****
835 REM *****
836 REM *****
837 REM *****
838 REM *****
839 REM *****
840 REM *****
841 REM *****
842 REM *****
843 REM *****
844 REM *****
845 REM *****
846 REM *****
847 REM *****
848 REM *****
849 REM *****
850 REM *****
851 REM *****
852 REM *****
853 REM *****
854 REM *****
855 REM *****
856 REM *****
857 REM *****
858 REM *****
859 REM *****
860 REM *****
861 REM *****
862 REM *****
863 REM *****
864 REM *****
865 REM *****
866 REM *****
867 REM *****
868 REM *****
869 REM *****
870 REM *****
871 REM *****
872 REM *****
873 REM *****
874 REM *****
875 REM *****
876 REM *****
877 REM *****
878 REM *****
879 REM *****
880 REM *****
881 REM *****
882 REM *****
883 REM *****
884 REM *****
885 REM *****
886 REM *****
887 REM *****
888 REM *****
889 REM *****
890 REM *****
891 REM *****
892 REM *****
893 REM *****
894 REM *****
895 REM *****
896 REM *****
897 REM *****
898 REM *****
899 REM *****
900 REM *****
901 REM *****
902 REM *****
903 REM *****
904 REM *****
905 REM *****
906 REM *****
907 REM *****
908 REM *****
909 REM *****
910 REM *****
911 REM *****
912 REM *****
913 REM *****
914 REM *****
915 REM *****
916 REM *****
917 REM *****
918 REM *****
919 REM *****
920 REM *****
921 REM *****
922 REM *****
923 REM *****
924 REM *****
925 REM *****
926 REM *****
927 REM *****
928 REM *****
929 REM *****
930 REM *****
931 REM *****
932 REM *****
933 REM *****
934 REM *****
935 REM *****
936 REM *****
937 REM *****
938 REM *****
939 REM *****
940 REM *****
941 REM *****
942 REM *****
943 REM *****
944 REM *****
945 REM *****
946 REM *****
947 REM *****
948 REM *****
949 REM *****
950 REM *****
951 REM *****
952 REM *****
953 REM *****
954 REM *****
955 REM *****
956 REM *****
957 REM *****
958 REM *****
959 REM *****
960 REM *****
961 REM *****
962 REM *****
963 REM *****
964 REM *****
965 REM *****
966 REM *****
967 REM *****
968 REM *****
969 REM *****
970 REM *****
971 REM *****
972 REM *****
973 REM *****
974 REM *****
975 REM *****
976 REM *****
977 REM *****
978 REM *****
979 REM *****
980 REM *****
981 REM *****
982 REM *****
983 REM *****
984 REM *****
985 REM *****
986 REM *****
987 REM *****
988 REM *****
989 REM *****
990 REM *****
991 REM *****
992 REM *****
993 REM *****
994 REM *****
995 REM *****
996 REM *****
997 REM *****
998 REM *****
999 REM *****
1000 REM *****

```

A) De avencer:

- 1) NORTE
- 2) SUR
- 3) ESTE
- 4) OESTE
- 5) ARRIBA
- 6) ABAJO

B) De acción:

- 1) COGE
- 2) DEJA
- 3) MATA
- 4) CORTA
- 5) NADA
- 6) ABRE
- 7) CAVA
- 8) COME
- 9) SALTA
- 10) SEBE
- 11) AYUDA
- 12) INVENTARIO
- 13) PUNTOS



Los mejores libros para tu SPECTRUM



digital, s.a.

PILAR DE ZARAGOZA, 45 (Seresquena a Cartago) - TELS. 246 40 90-246 56 03 - 28028 MADRID - APDO. CORREDS 6287

CODIGO	TITULO	P.V.P.
2307004	SPECTRUM INTRODUCCION AL MUNDO MAQUINA Para aquellos que no se encuentran con el. Fácil y directo aprender todo desde el primer momento.	1.400 Ptas.
2307004	EL PROGRAMADOR DE SPECTRUM Este es el libro para todos los usuarios de Spectrum que quieren aprender o ampliar sus propios programas.	1.300 Ptas.
2493300	SPECTRUM LIBRO DE JUEGOS Colección de juegos para los ZX Spectrum con instrucciones para mejorar su programación.	1.200 Ptas.
3021801	PROGRAMACION PASO A PASO CON EL SPECTRUM Este funcionamiento para cada el perfil posible a lo largo de 4 y 6 niveles de dificultad que se van en el futuro.	700 Ptas.
3021804	LAS PRIMERAS 5 IDEAS CON EL SPECTRUM Una selección para que el mismo usuario durante los primeros momentos de aprendizaje de.	700 Ptas.
2493302	ZX SPECTRUM COMO COMENZAR EL MAXIMO BENEFICIO Si algún día en el futuro se quiere a conocer su ordenador que puede programar en Spectrum para que haya el que sea.	1.200 Ptas.
2000102	COMO USAR LOS COLORES Y LOS GRAFICOS EN EL SPECTRUM Introducción al uso de gráficos en el mundo de Spectrum que permite el usuario de probar desde el inicio y avanzando.	600 Ptas.
2000101	COMO PROGRAMAR EN SPECTRUM Curso de programación de los usuarios de Spectrum que permite el usuario de probar desde el inicio y avanzando.	600 Ptas.
2307005	50 JUEGOS EDUCATIVOS PARA EL SPECTRUM Colección de programas para facilitar el aprendizaje de los usuarios de Spectrum y divertirse a la vez.	1.200 Ptas.
2307200	OBTENER EL MAXIMO BENEFICIO A SU ZX SPECTRUM Una colección de los que en el mundo de Spectrum y dentro de los usuarios que quieren el mundo de los usuarios de Spectrum.	1.000 Ptas.
2307201	GRAFICOS Y SONIDOS EN EL SPECTRUM Un curso para aquellos que quieren aprender a programar el dibujo y los sonidos en el mundo de Spectrum.	1.000 Ptas.
2493303	ZX SPECTRUM INTRODUCCION AL PROGRAMADOR DE TEXTOS Este libro contiene muchas de programación y algunos programas como sea para que el Spectrum sea el primer programa de texto.	1.000 Ptas.
2307006	COMO MAQUINA ZX SPECTRUM Para los usuarios que quieren conocer los detalles de la máquina de Spectrum y cómo se utiliza.	2.000 Ptas.
2307008	LOS 20 MEJORES PROGRAMAS PARA EL ZX SPECTRUM Este libro presenta una selección de los mejores programas en el mundo de Spectrum.	1.000 Ptas.
2307009	PROGRAMACION AVANZADA PARA EL ZX SPECTRUM Este libro presenta una selección de los mejores programas en el mundo de Spectrum.	1.000 Ptas.
2307010	LAS 10 MEJORES SUBROUTINAS Colección de los mejores programas en el mundo de Spectrum.	1.000 Ptas.
2493304	ZX SPECTRUM: TEORIA Y PRACTICA DE PROGRAMACION Este libro presenta una selección de los mejores programas en el mundo de Spectrum.	1.000 Ptas.

ENVIAR A: DIGITAL, S.A. C/ Pilar de Zaragoza, 45 28028 MADRID

Apellidos:
 Nombre:
 Dirección:
 Localidad: D.P.
 Provincia: Teléfono:

CUPON DE PEDIDO		CANTIDAD	PRECIO
REFERENCIA	TITULO		

Forma de pago: Contra reembolso GASTOS DE ENVIO 200
 Tarjeta de crédito: Visa Eurocard TOTAL

Nº de tarjeta:
 Fecha de caducidad:

FIRMA:
 Irrevocable para pago con tarjeta



RELLENAR A MAQUINA O CON MAYUSCULAS

RELLENAR A MAQUINA O CON MAYUSCULAS